Package ‘ABHgenotypeR’

August 29, 2016

Type Package
Title Easy Visualization of ABH Genotypes
Version 1.0.1
Date 2016-02-04
Description Easy to use functions to visualize marker data from biparental populations. Useful for both analyzing and presenting genotypes in the ABH format.
License GPL-3
URL http://github.com/StefanReuscher/ABHgenotypeR
BugReports http://github.com/StefanReuscher/ABHgenotypeR/issues
LazyData TRUE
Imports ggplot2, reshape2, utils
RoxygenNote 5.0.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Stefan Reuscher [aut, cre], Tomoyuki Furuta [aut]
Maintainer Stefan Reuscher <reuscher.stefan@gmail.com>
Repository CRAN
Date/Publication 2016-02-04 11:27:29

R topics documented:

ABHgenotypeR ................................................................. 2
correctStretches .......................................................... 2
correctUndercalledHets ................................................ 3
imputeByFlanks ............................................................ 3
plotAlleleFreq .............................................................. 4
correctStretches

Description
Correct short miscalled stretches based on flanking alleles.

Usage
```r
correctStretches(inputGenos = "genotypes", maxHapLength = 1)
```

Arguments

- `inputGenos` A genotypes list object.
- `maxHapLength` The maximum length of stretches flanked by non-heterzygous sites that are changed. If set to 1 (default) only AXA or BXB will be corrected. If set to 2, both AXA and AXYA (or BXB and BXYB) will be corrected.

Value
A genotype object in which short miscalled stretches are corrected if both flanking alleles match.

Examples
```r
## Not run: corrStretchGenos <- correctStretches(genotypes, maxHapLength = 3)
```
**correctUndercalledHets**

*Correct undercalled heterozygous sites based on flanking alleles.*

**Description**

Correct undercalled heterozygous sites based on flanking alleles.

**Usage**

```r
correctUndercalledHets(inputGenos = "genotypes", maxHapLength = 1)
```

**Arguments**

- `inputGenos` : A genotypes list object.
- `maxHapLength` : The maximum length of not heterozygous stretches flanked by heterzygous sites that are changed to heterozygous. If set to 1 (default) only HAH or HBH will be corrected. If set to 2, both HAH and HAAH (or HBH and HBBH) will be corrected.

**Value**

A genotype object in which undercalled heterozygous sites are corrected if both flanking alleles match.

**Examples**

```R
# Not run: corrUndHetsGenos <- correctUndercalledHets(genotypes, maxHapLength = 3)
```

**imputeByFlanks**

*Impute missing genotypes based on flanking alleles*

**Description**

Impute missing genotypes based on flanking alleles.

**Usage**

```r
imputeByFlanks(inputGenos = "genotypes")
```

**Arguments**

- `inputGenos` : A genotypes list object.
Value
A genotype object in which missing data is imputed based on flanking alleles. Any number of N is replaced by either A, B or N if the alleles which flank the N match.

Examples
```r
## Not run: imputedGenos <- imputeByFlanks(genotypes)
```

---

**plotCompareGenos**

Description
Compare to genotype matrices

Usage
```r
plotCompareGenos(genos_1 = "genotypes_1", genos_2 = "genotypes_2",
markerToPlot = "all", individualsToPlot = "all", chromToPlot = "all",
CompColors = c("#000000", "#E6F00"), textSize = 12,
showMarkerNames = FALSE, showIndividualNames = FALSE)
```

---

**plotAlleleFreq**

Plot the parental allele frequencies along the chromosomes.

Description
Plot the parental allele frequencies along the chromosomes.

Usage
```r
plotAlleleFreq(genos = "genotypes")
```

Arguments
- genos: The output of readABH genotypes

Value
A plot of parental allele frequencies along the chromosomes. If the output is assigned a name a ggplot2 object is returned for further manipulation.

Examples
```r
## Not run: plotAlleleFreq(genotypes)
## Not run: p <- plotAlleleFreq(genotypes)
```
Arguments

- **genos_1**: Output of `readABHgenotypes`
- **genos_2**: Output of `readABHgenotypes`. Note that both genos object need to have identical numbers of marker x individuals.
- **markerToPlot**: A character vector of marker names which appear in the plot. Defaults to all.
- **individualsToPlot**: A character vector of individual names which appear in the plot. Defaults to all.
- **chromToPlot**: A character vector of chromosome names which appear in the plot. Defaults to all.
- **CompColors**: A character vector of length 2 giving the color names or values to use for different and identical markers. Defaults to black and orange.
- **textSize**: The size of all text elements in the plot. Useful for making a nice plot. Defaults to 12.
- **showMarkerNames**: Show the marker names along the x axis. This and showIndividualnames are useful when you display only a few markers and want them labeled. Defaults to FALSE.
- **showIndividualNames**: Show individual names along the y axis.

Value

A graphical comparison of genotypes.

Examples

```r
## Not run: plotCompareGenos(preImpGenotypes, postImpGenotypes)
## Not run: #for more examples see plotGenos()
```

Description

Plot graphical genotypes.

Usage

```r
plotGenos(genos = "genotypes", markerToPlot = "all", individualsToPlot = "all", chromToPlot = "all", alleleColors = c("#56B4E9", "#E69F00", "#009E73", "#000000"), textSize = 12, showMarkerNames = FALSE, showIndividualNames = FALSE)
```
### Arguments

- **genos**
  - The output of `readABHgenotypes`
- **markerToPlot**
  - A character vector of marker names which appear in the plot. Defaults to all.
- **individualsToPlot**
  - A character vector of individual names which appear in the plot. Defaults to all.
- **chromToPlot**
  - A character vector of chromosome names which appear in the plot. Defaults to all.
- **alleleColors**
  - A character vector of length 4 giving the color names or values to use for the A,B,H and n.d genotypes. Defaults to orange, blue, green and black.
- **textSize**
  - The size of all text elements in the plot. Useful for making a nice plot. Defaults to 12.
- **showMarkerNames**
  - Show the marker names along the x axis. This and `showIndividualNames` are useful when you display only a few markers and want them labeled. Defaults to `FALSE`.
- **showIndividualNames**
  - Show individual names along the y axis.

### Value

Graphical genotypes.

### Examples

```r
## Not run: plotGenos(genotypes)
makerNames <- c("marker1", "marker2", "marker3")
individualNames <- c("F2_100", "F2_101", "F2_102", "F2_103")
someColors <- c("black", "red", "gold", "white")
## Not run: plotGenos(genotypes, makerNames, individualNames, 1:3, someColors)
## Not run: p <- plotGenos(genotypes)
```

---

**plotMarkerDensity**

Plot the marker density along the chromosomes.

### Description

Plot the marker density along the chromosomes.

### Usage

```r
plotMarkerDensity(genos = "genotypes")
```

### Arguments

- **genos**
  - The output of `readABHgenotypes`
**readABHgenotypes**

**Value**

A plot of marker densities along the chromosomes. If the output is assigned a name a ggplot2 object is returned for further manipulation.

**Examples**

```r
## Not run: plotMarkerDensity(genotypes)
## Not run: p <- plotMarkerDensity(genotypes)
```

---

**Description**

Read in the output of the genosToABH plugin.

**Usage**

```r
readABHgenotypes(pathToABH, nameA = "A", nameB = "B", readPos = TRUE)
```

**Arguments**

- `pathToABH` The path and filename of the input file.
- `nameA` Name of the parent represented by "A" in the input file.
- `nameB` Name of the parent represented by "B" in the input file.
- `readPos` Should the function attempt read the physical position of markers from the input?

**Details**

The input files should be a .csv file holding genotypes as specified by the qtl package and its "csvs" format. All characters in the genotype matrix which are not either A,B or H will be set to N. If `readPos = TRUE` (default) marker names must conform to S1_123456 meaning 123456 bp on chromosome 1. If FALSE, pos is set to NULL and needs to be manually constructed as shown in the examples. Note that this might throw off some plotting function.

**Value**

A genotype list object which holds the information from the input file. This list is the fundamental datastructure used by the other functions in this package. See the vignette for what each item in the list is.

**Examples**

```r
## Not run: genotypes <- readABHgenotypes("./genotypes.csv", "NB", "QL")

## Not run: otherGenotypes <- readABHgenotypes("./otherGenotypes.csv", readPos = FALSE)
#arbitrary position to keep marker order intact
## Not run: otherGenotypes$pos <- 1:length(otherGenotypes$marker_names)
```
**reportGenos**

*Report the total and relative number of each allele in a genotype object.*

**Description**

Report the total and relative number of each allele in a genotype object.

**Usage**

```
reportGenos(genos1)
```

**Arguments**

- `genos1`: A genotypes list object.

**Value**

Console output of the total an relative number of each allele.

**Examples**

```r
## Not run: reportGenos(preImputation)
```

---

**writeABHgenotypes**

*Export a genotype list to a .csv file.*

**Description**

Export a genotype list to a .csv file.

**Usage**

```
writeABHgenotypes(genos = "genotypes", outfile = "./outfile.csv")
```

**Arguments**

- `genos`: The output of readABHgenotypes or one of the imputation/error correction functions.
- `outfile`: The path and filename of the output file.

**Value**

A file which can be used in R/qtl or elsewhere.

**Examples**

```r
## Not run: writeABHgenotypes(genotypes, outfile = "./outfile_name.csv")
```
Index

ABHgenotypeR, 2
ABHgenotypeR-package (ABHgenotypeR), 2

correctStretches, 2
correctUndercalledHets, 3

imputeByFlanks, 3

plotAlleleFreq, 4
plotCompareGenos, 4
plotGenos, 5
plotMarkerDensity, 6

readABHgenotypes, 7
reportGenos, 8

writeABHgenotypes, 8